INDEX

SECTION 1

SECTION 2

SECTION 3

SECTION 4

SECTION 1 EXTENSION 1

1a/ INTRODUCTION

N.M.I. COMPUTERS' EXTENSION 1 program is contained in one 2708 type E.P.R.O.M. This 1K extension is not a piece of general purpose Z80 software and is designed to work exclusively with NAS-SYS 1 or 3 in any NASCOM 1 or 2 based machine.

The program provides 20 extra useful routines, called in the same way as NAS-SYS (see Section 1f), to enable the user to develop his more complex programs, particularly in the fields of number handling and graphics, more easily and quickly.

1b/ LOCATION

In its standard form EXTENSION 1 occupies memory space from B000H to B3FFH although if required it can be supplied originating on any 4K boundary.

Because it is housed in a 2708 type E.P.R.O.M. it is suitable for plugging into any one of the 8 1K ROM/RAM sockets on a NASCOM 2. For connection of the standard E.P.R.O.M. see Section 1i. It should be noted that the remainder of the 4K block should be left empty to ensure correct initialization of EXTENSION 1 and to allow for expansion.

1c/ SERIAL NUMBER

Each copy of EXTENSION 1 has a unique serial number which, on initialization, will be displayed on the right hand side of the unscrolled top line. It will remain there along with the heading message until written over by a user program or a CLEAR SCREEN is issued.

1d/ WORKSPACE

EXTENSION 1 requires some RAM to store the jump table. These locations are from 0C80H to 0D26H inclusive.

On initialization EXTENSION 1 copies the NAS-SYS jump table from R.O.M. to locations 0C82H to 0CFFH then copies its own jump table from R.O.M. to locations 0D00H to 0D26H.

The R.A.M. locations 0C71H and 0C72H in NAS-SYS workspace are also modified to point to the start of the new jump table, i.e. 0C00H.

Because NAS-SYS 1 is 3 routines shorter than NAS-SYS 3, the spare locations are filled, by EXTENSION 1, with vectors to error messages. This action prevents the user losing control of the machine by executing an invalid subroutine call.

1e/ USING EXTENSION 1 WITH OTHER PROGRAMS

If the user has any existing programs that reouire R.A.M. at the locations mentioned in Section 1d then EXTENSION 1 must not be initialized if control of the machine is to be maintained.

Some users may be in the habit of writing small programs starting at locations between ØC80H and the estimated bottom of the stack. Again if EXTENSION 1 is to be used and control maintained then it is recommended that programs be øriginated at 1000H.

Programs such as ZEAP and NAS-DIS are quite happy operating with NAS-SYS in conjunction with EXTENSION 1 although depending upon the origin of your copy of EXTENSION 1 and space required by other software some care may have to be exercised in the specification of memory areas.

1f/ CALLING EXTENSION 1 ROUTINES

To call any NAS-SYS routine the op-code DF is typed in, followed by a single byte routine number. The routine number is used to generate an offset into the jump table whose starting address is stored at OC71H and OC72H.

EXTENSION 1 routines are called in the same way, just follow the DF with a routine number between 80H and 93H.

For the routines and their routine numbers, see Section 2.

1g/ INITIALIZATION OF EXTENSION 1

Users of EXTENSION 1 at its standard address who possess NAS-SYS-3 need only type 'Y' then 'Enter'. If, however, your system differs from the above then simply type 'EB000', or whatever your address is, followed by 'Enter'.

The machine will respond by writing the following message to the unscrolled line:-

```
NAS-SYS X NMI COMPUTERS Extension 1 XXXX
```

The cursor will also change to ':' to signify that EXTENSION 1 routines are available for use.

1h/ DE-INITIALIZATION OF EXTENSION 1

The simplest way of effecting this is to issue a RESET. However, this may cause some system parameters to be corrupted so it is recommended that locations OC71H and OC72H returned to their original values using the NAS-SYS MODIFY command.

1i/ INSTALLING OF EXTENSION 1

This section describes the recommended method of installing EXTENSION 1 at its standard address.

The R.O.M. should be inserted into the I.C. socket marked A1 or I.C. 35, just to the left of BASIC if PL1 is to the RIGHT.

To address EXTENSION 1 at BOOOH, additional links should be soldered to the MEMORY DECODE LINK BLOCK LKS1 between pins 12, 4 and 7.

SECTION 2

SUMMARY OF ROUTINES

ROUTINE NO.	NAME	ARGUMENT REGISTERS	REGISTERS CORRUPTED
80	SUMLT	H x L → HL	-
81	SSMLT	/H x L → HL	-
82	DUMLT	HL x DE → HLDE	-
83	DSMLT	HL x DE → HLDE	F
84	SUDIV	H/L → H	Z set if /0
85	SSDIV	H/L → H	Z set if /0
86	DUDIV	HL/DE → HL	Z set if /0
87	DSDIV	HL/DE → HL	Z set if /0
88	SSFDV	(H/L) x 256 → HL	Z set if /0
89	STZFG	SET ZERO FLAG	-
8A	RSZFG	RESET ZERO FLAG	-
8B	TCMHL	TWO'S COMP HL	-
8C	TCMDE	TWO'S COMP DE	-
8D	TCMBC	TWO'S COMP BC	-
8E	THLDE	TWO'S COMP HLDE	-
8F	SETPX	*SET PIXEL in DE X in D, Y in E	A Z SET IF OUT OF RANGE
90	RESPX	*RESET PIXEL in DE X in D, Y in E	A Z SET IF OUT OF RANGE
91	POINT	*POINT TO PIX in DE (C set if set)	A Z SET IF OUT OF RANGE
92	DLINE	*LINE FROM HL → DE	-
93	SHAPE	*SHAPE DRAW	A

* ARGUMENTS IN HEXADECIMAL

SECTION 3

EXPLANATION OF ROUTINES

3a/ ARITHMETIC ROUTINES

Routines with numbers between 80H and 88H are arthmetically oriented. The use of these routines can be best understood by examining the ROUTINE SUMMARY in SECTION 2 and SECTION 4.

One point worthy of note however, is an attempt to divide by zero will cause the routine to return with the zero flag set.

3b/ SET ZERO FLAG (STZFG) DF 89

This routine, when called, will cause the ZERO FLAG in the Z80 'F' register to be set to a logic 1.

3c/ RESET ZERO FLAG (RSZFG) DF 8A

This routine, when called, will cause the ZERO FLAG in the Z80 'F' register to be reset to a logic 0.

3d/ TWO'S COMPLEMENT REGISTER PAIRS DF 8B, DF 8C, DF 8D

These routines will perform a two's complementation on register pairs HL, DE, BC respectively. For examples of their use refer to SECTION 4.

3e/ TWO'S COMPLEMENT REGISTER PAIRS HL and DE DF 8E

For this routine register pairs HL and DE are cascaded forming a 'register quad'. Register pair HL form the most significant word and register pair DE the least significant word.

For an example of its use refer to SECTION 4

3f/ SET (LIGHT UP) SCREEN PIXEL DF 8F

RESET (EXTINGUISH) SCREEN PIXEL DF 90

POINT (TEST) SCREEN PIXEL DF 91

These three routines are similar to the routines provided by NASCOM BASIC. The NASCOM screen is divided into a grid 96 points wide and 45 points high. The Pixels are numbered 0-95 on the X axis & 0-44 on the Y axis.

To address a particular pixel load register pair DE with the pixel co-ordinates then call the routine required.

The co-ordinates are loaded into DE in hexadecimal, register D holding the 'X' co-ordinate and register E holding the 'Y' co-ordinate. So, for example, to address pixel 10, 20 simply load DE with OA, 14.

When testing the state of a pixel the routine will return with the 'CARRY FLAG' set to a logic 1 if the pixel is lit.

All these routines corrupt the Z80 'A' register.

3g/ LINE DRAW (D LINE) DF 92

This routine also uses the NASCOM 96 x 45 screen to draw a straight line between any two points on the grid.

Register pair HL holds X1 and Y1 respectively and register pair DE holds X2 and Y2 respectively.

Note that owing to the Z80 structure the arguments are loaded in backwards i.e. the low byte (Y) first then the high byte (X).

To save time the arguments are also loaded in hexadecimal, the conversion from the decimal screen to argument value being done by the user.

Referring to SECTION 4 part 17 will show a program to draw a line from point OH, OH to 30H, 20H or O, O to 48, 32.

3h/ SHAPE DRAW (SHAPE) DF 93

This routine differs from the others in its format.

The routine is called by first entering the routine call i.e. DF 93, then the number of arguments the routine it to expect (127 max.) followed by the arguments themselves.

If more than one shape is required, to save time and space repeatedly calling SHAPE, the facility of "moving without drawing" has been included, this being selected by setting the M.S.B. of the destination co-ordinates. (See SECTION 4 part 18).

The arguments used in this routine are all entered in hexadecimal as in D LINE. Register 'A' is corrupted by this routine.

SECTION 4 EXAMPLES OF USE OF EXTENSION 1 ROUTINES

The following short programs are given here to illustrate the use of EXTENSION 1 routines.

They should be typed in using the NAS-SYS MODIFY command starting at location 1000H. They can then be executed by typing E1000 then 'ENTER'.

1/ SINGLE PRECISION UNSIGNED MULTIPLICATION (SUMLT) DF 80

```
M1000
 1000        26         05         LD H 5
 1002        2E         OA         LD L 10
 1004        DF         80         CALL SUMLT
 1006        DF         66         PRINT HL
 1008        DF         5B         MRET

E1000
0032 or 50 in decimal.
```

2/ SINGLE PRECISION SIGNED MULTIPLICATION (SSMLT) DF 81

```
M1000
 1000        26         FB         LDH -5
 1002        2E         OA         LDL 10
 1004        DF         81         CALL SSMLT
 1006        DF         66         PRINT HL
 1008        DF         5B         MRET

E1000
FFCE or -50 in decimal.
```

3/ DOUBLE PRECISION UNSIGNED MULTIPLICATION (DUMLT) DF 82

```
M1000
 1000        21         64         00          LD HL 100
 1003        11         32         00          LD DE  50
 1006        DF         82                     CALL DUMLT
 1008        DF         66                     PRINT HL
 100A        EB                                EX DE HL
 100B        DF         66                     PRINT 'DE'
 100D        DF         5B                     MRET

E1000
0000 1388 or 5000 in decimal.
```

4/ DOUBLE PRECISION SIGNED MULTIPLICATION (DSMLT) DF 83

```
M1000
 1000      21       9C       FF       LD HL -100
 1003      11       32       00       LD DE   50
 1006      DF       83                CALL DSMLT
 1008      DF       66                PRINT HL
 100A      EB                         EX DE HL
 100B      DF       66                PRINT 'DE'
 100D      DF       5B                MRET

E1000
FFFF EC78 or -5000 in decimal.
```

5/ SINGLE PRECISION UNSIGNED DIVISION (SUDIV) DF 84

```
M1000
 1000      26       64       LD H 100
 1002      2E       05       LD L 5
 1004      DF       84       CALL SUDIV
 1006      7C                LDA H
 1007      DF       67       PRINT A
 1009      DF       5B       MRET

E1000
14 or 20 in decimal.
```

6/ SINGLE PRECISION SIGNED DIVISION (SSDIV) DF 85

```
M1000
 1000      26       9C       LD H -100
 1002      2E       05       LD L 5
 1004      DF       85       CALL SSDIV
 1006      7C                LDA H
 1007      DF       67       PRINT A
 1009      DF       5B       MRET

E1000
EC or -20 in decimal.
```

7/ DOUBLE PRECISION UNSIGNED DIVISION (DUDIV) DF 86

```
M1000
 1000      21      10      27      LD HL 10,000
 1003      11      14      00      LD DE 20
 1006      DF      86              CALL DUDIV
 1008      DF      66              PRINT HL
 100A      DF      5B              MRET

E1000
01F4 or 500 in decimal.
```

8/ DOUBLE PRECISION SIGNED DIVISION (DSDIV) DF 87

```
M1000
 1000      21      FO      DB      LD HL -10,000
 1003      11      14      00      LD DE 20
 1006      DF      87              CALL DSDIV
 1008      DF      66              PRINT HL
 100A      DF      5B              MRET

E1000
FEOC or -500 in decimal.
```

9/ SINGLE PRECISION SIGNED FRACTIONAL DIVISION (SSFDV) DF 88

```
M1000
 1000      26      OF      LD H 15
 1002      2E      02      LD L 2
 1004      DF      88      CALL SSFDV
 1006      DF      66      PRINT HL
 1008      DF      5B      MRET

E1000
0780 or 7.5 in decimal.
```

10/ TWO'S COMPLEMENT REGISTER PAIR HL (TCMHL) DF 8B

```
M1000
 1000      21      64      00      LD HL 100
 1003      DF      8B              CALL TCMHL
 1005      DF      66              PRINT HL
 1007      DF      5B              MRET

E1000
FF9C or -100 in decimal.
```

11/ TWO'S COMPLEMENT REGISTER PAIR DE (TCMDE) DF 8C

```
M1000
 1000      11      64      00      LD DE 100
 1003      DF      8C              CALL TCMDE
 1005      EB                      EX DE HL
 1006      DF      66              PRINT 'DE'
 1008      DF      5B              MRET

E1000
FF9C or -100 in decimal.
```

12/ TWO'S COMPLEMENT REGISTER PAIR BC (TCMBC) DF 8D

```
M1000
 1000      01      64      00      LD BC 100
 1003      DF      8D              CALL TCMBC
 1005      C5                      PUSH BC
 1006      E1                      POP HL
 1007      DF      66              PRINT 'BC'
 1009      DF      5B              MRET

E1000
FF9C or -100 in decimal.
```

13/ TWO'S COMPLEMENT REGISTER PAIRS HL DE (THLDE) DF 8E

```
M1000
 1000      21      00      00      LD HL 0
 1003      11      64      00      LD DE 100
 1006      DF      8E              CALL THLDE
 1008      DF      66              PRINT HL
 100A      EB                      EX DE HL
 100B      DF      66              PRINT 'DE'
 100D      DF      5B              MRET

E1000
FFFF FF9C or -100 in decimal.
```

14/ 'SET' (LIGHT UP) SCREEN PIXEL (SETPX) DF 8F

```
M1000
 1000      16      20      LD D 32 (X)
 1002      1E      20      LD E 32 (Y)
 1004      DF      8F      CALL SETPX
 1006      DF      5B      MRET

E1000
```

15/ 'RESET' (EXTINGUISH) SCREEN PIXEL (RESPX) DF 90

```
M1000
 1000      16      20      LD D 32 (X)
 1002      1E      20      LD E 32 (Y)
 1004      DF      90      CALL RESPX
 1006      DF      5B      MRET

E1000
```

16/ POINT TO (TEST) SCREEN PIXEL (POINT) DF 91

```
M1000
 1000      16      20      LD D 32 (X)
 1002      1E      20      LD E 32 (Y)
 1004      DF      91      CALL POINT (CARRY FLAG SET IF LIT)
 1006      DF      5B      MRET

E1000
```

17/ DRAW LINE FROM X1, Y1 to X2, Y2 (D LINE) DF 92

```
M1000
 1000        21        00*        00*        LD HL 00, 00 (X1, Y1)
 1003        11        20*        30*        LD DE 48, 32 (X2, Y2)
 1006        DF        92                    CALL D LINE
 1008        DF        5B                    MRET

E1000
```

* NOTE ARGUMENTS ARE REVERSED DUE TO Z80 STRUCTURE

i.e. Low byte first, then high byte.

This program will draw a line from screen point 0, 0 to 48, 32

18/ SHAPE DRAW (SHAPE) DF 93

```
M1000
 1000        DF        93        CALL SHAPE
 1002        0A                  NUMBER OF ARGUMENTS IN HEX (10)
 1003        0A        0A        FIRST SET OF CO-ORDS (10, 10)
 1005        40        0A        64, 10
 1007        40        28        64, 40
 1009        0A        28        10, 40
 100B        0A        0A        10, 10
 100D        90        10        16, 16 (MSB SET so line not drawn)
 100F        3A        10        58, 16
 1011        3A        22        58, 34
 1013        10        22        16, 34
 1015        10        10        16, 16
 1017        DF        5B        MRET
```

This program will draw two rectangles, one inside the other.

MB/CJM

23/12/82